

文章编号: 1000-582X(2011)12-109-06

WNXD 数据库存储策略的研究和实现

刘荷花

(太原大学 计算机系, 山西 太原 030009)

摘要: 针对 NativeXML 数据存储性能不高、支持查询和更新效率低等不足, 提出一种新的半结构化信息存储模型 WNXD。引入动态倒排技术和数据映像机制, 将记录和分页的存储模式相结合, 建立 3 种结构化索引, 用不同方法解决了结构化和动态增长的矛盾。创建了实用的存储模型, 提供了从整体到局部, 从元素、属性到具体值的全面覆盖访问策略, 底层的数据存取, 尤其是内容混杂的各类形式异构数据的有效性和完整性存取, 得到了最大程度保证, 查询次数减小了 I/O, 提高了系统处理性能。

关键词: WNXD; 数据存储; 数据查询; 数据更新

中图分类号: TP392

文献标志码: A

Research and implementation on WNXD database storage strategy

LIU He-hua

(School of Computer, Taiyuan University, Taiyuan 030009, Shanxi, P. R. China)

Abstract: For the drawbacks of NativeXML, such as low data storage performance, low efficiency of querying and updating, a new kind of semi-structured information storage model, named WNXD, is proposed. The dynamic invert-sorting technique and data imaging mechanism are introduced to combine the record and the page storage mode, three structured indexes are established to achieve structural and dynamic growth with different ways. A practical storage mode is created to provide a comprehensive access strategy from global to local, and from elements, attributes to values, which guarantees effective and complete data storage, especially for the data with different types. The query number decreases I/O and the system performance is improved.

Key words: WNXD; data storage; data query; data update

XML (extensible markup language) 是一种标记语言, 成为一种网上通用的数据存储与信息交换的标准格式。高效管理 XML 数据成为当前计算机领域关注的一个重要问题。

1 XML 文档的缺陷

XML 是标准通用标记语言 SGML (stander generalized markup language) 的简化子集, 形式描

述与内容描述相分离, 数据结构具有开放性、自描述性、无限嵌套、树形结构等特性, 被广泛应用于网络信息检索和标准化文档传输, 但由于 XML 的半结构化特点, 影响存储和查询处理的效率。

XML 的基本语法有如下规定: 有关版本的声明, 可以作为文件的第一条语句; 标记语法的各规则, 在标记的使用时, 要加以遵循; 用双引号囊括属性的值; 使用 XML 中特定的编码, 来表示特殊

收稿日期: 2011-07-03

基金项目: 国家自然科学基金资助项目(60475022); 山西省科技厅软科学资助项目(2010041016)

作者简介: 刘荷花(1976-), 女, 太原大学副教授, 主要从事计算机数据库方向研究, (Tel)13363511719;
(E-mail)liuhehua1999@126.com。

字符^[3]。

目前 XML 文档的主要存储策略有:面向对象数据库、关系数据库、原生 XML 数据库、Native XML 数据库等 4 种。

面向对象数据库:利用对象查询语言(OQL)实现对 XML 数据的结构化查询;支持自定义数据类型;使用父子关系描述数据关系;数据模型是层次树。对于不符合模式的数据须单独处理,查询效率较低,代价较大。

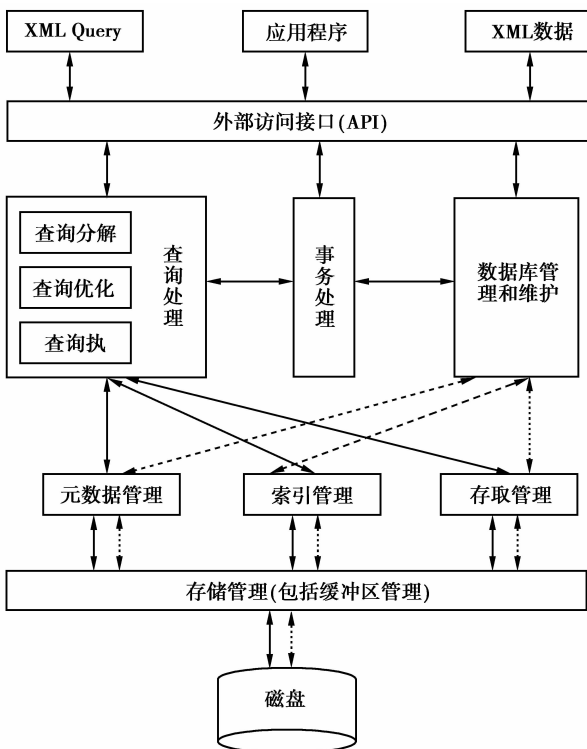


图 1 Native XML 数据库系统体系结构

关系数据库:是完全结构化的模型,表达的信息非常规则,把 XML 文档分割映射成多个相互关联的关系数据表,转换成 SQL 片段。对于非规则数据存储效率较低。

原生 XML 数据库:存储数据时原有数据不轻易移植到新系统中,不进行 XML 和数据库的映射转换,不注重数据的底层存储结构,只维持原有 XML 文档的数据结构和相关元数据,目前技术还不够成熟。

Native XML 数据库:不取代传统的数据库,结构多种多样,内部模型保持 XML 的树形结构和相关元数据,以自然方式来处理数据,适于存储和查询“以文档为中心”的 XML 数据,但存储性能不高,支持查询和更新不很高效。

理想存储结构的关键:在于设计出存储文件结

构和索引结构。

2 WNXD 的体系结构

设计了一种新的数据库—WNXD,W 是 well 的缩写,代表“好”,NXD 是 Native XML DataBase 的缩写,构建了数据文件结构和索引文件结构^[2-3],优化了以往的 NXD 存储方法。WNXD 的体系结构包括 API、数据管理引擎、解析器、数据库管理、存储区。如图 2 所示。

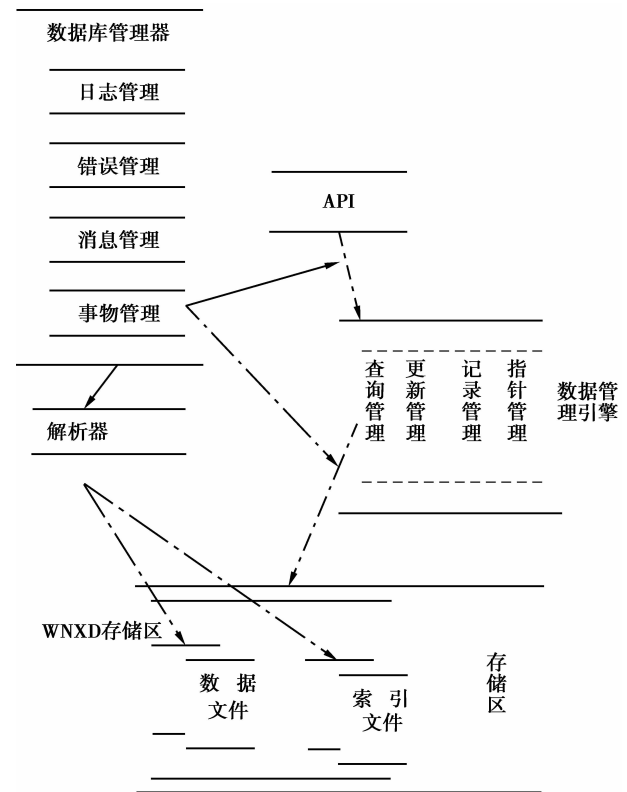


图 2 WNXD 的体系结构

1) API:按照标准化要求,依据服务、资源、集合、驱动等功能创建。

2)管理数据引擎:文件的查询、引入、更新,使用数据库驱动列表完成。

3)解析器:主要负责建立数据库,如文件头写法、文件体内容等。

4)数据库管理:负责日志管理、错误管理、消息管理、事务管理等,使数据库运行正常。

5)存储区域:按照 WNXD 规范,生成索引等数据文件,加以存储。

3 WNXD 数据存储方案设计

3.1 WNXD 文件存储

文件存储主要有:顺序文件,随机文件和二进制

文件 3 种(如图 3 所示)。采用二进制存储 WNXD 文件^[4-6]。

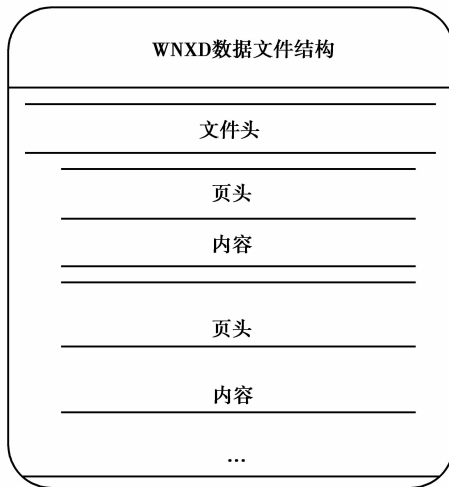


图 3 WNXD 数据文件结构

WNXD 数据文件:由数据内容(record data)、页内索引(record index in page)、文件头(fileheader)3 部分组成。文件头:由版本号、数据库文件大小、创建时间等信息组成。页内索引:由页号、开始顺序号和开始“层”数组成。文件内容由数据记录表和原数据组成。使用数据记录表正向存储、原数据从本页最后逆向存储的正反存储的方法,可以克服数据记录表存储空间设置大浪费空间、设置小不够存储的不足。

WNXD 索引文件:由地址表、聚簇表组成。文件和文件集存储在数据库中,地址表记录数据文件的动态存储地址,可以快速找到每一个数据文件。聚簇表是一个元素倒排表,为数据压缩存储提供前提,数据库中的属性名,或任意一个标签名,可以通过聚簇表快速找到^[7-8]。

3.2 WNXD 存储技术

WNXD 的数据是顺序存储的^[2]。引入“层”的概念,采取逐条解析分页存储,数据文件和索引文件相分离的存储策略。引入的外部索引和页结构,尤其是“层”的概念的引入,支持了 Native XML 数据库对更新的操作。对增量更新的操作,支持更为明显。

3.2.1 “层”的原理和技术

一些很难的问题,引入“层”后,得以方便解决。特别便于确定祖先子孙关系、父子关系、兄弟关系。

如:假定文档树为 A,节点为 a:

祖先子孙关系的确定:通过确定 a 的层数,来

确定 a 子孙节点的范围。如果节点 b 与 a 同层,并且大于 a,则 a 的子孙,为 a 和 b 之间的所有节点。同层节点如果小于 a,则 a 的子孙,为所有大于 a 的节点。

父子关系的确定:a 的下一层,a 的字节节点为所有子孙节点。

兄弟关系的确定:兄弟关系要符合两个条件:在同一层次、有相同的父亲节点,亲兄弟关系要加上父子关系,表兄弟关系需找到同层的节点。

3.2.2 逐条解析分页存储

逐条解析任意大小的 XML 文档数据,直到文档结束,当文件大于一定字节数时,分成许多“页”,进行分页存储,每一页大小固定,具有独立性,能快速寻址,方便操作(如:更新,修改)。

3.2.3 数据文件和索引文件分离

包括序言、尾声、注释等在内的所有 XML 数据(甚至看似无用的空格),按照 XML 文档的书写格式,通过 SAX API 解析,化为二进制格式,建立内部索引和外部索引文件。通过分离存储的方式,使每个文件自由增加长度更加有利,磁盘空间的利用更好、上层索引速度更快。

3.3 WNXD 存储数据流程

XML 文档解析器,用 SAX API 来作。读取 XML 文档数据时,依据解析器接口规则。按照 WNXD 规定的规则,将数据序列化,实现顺序无回溯高效存储。

WNXD 数据存储的流程图如图 4 所示。

3.4 WNXD 为增量更新提供一种 API 方案

已知:1)WNXD 存储中记录结束符号和记录开始符号放在一起。

2)插入 XML 数据,形式为: $\langle A \rangle C \langle /A \rangle$; ,是封闭的。

3)插入位置:层用 x 表示,顺序号用 Y 表示,在 (x_1, y_1) 和 (x_2, y_2) 之间,

插入 DATAS $\langle \langle A \rangle C \langle /A \rangle \rangle$

被更新页,根据 x_1, y_1 的值来确定。 x_1, x_2, y_1, y_2 间的关系,依据下列情况确定。 $x_1 = x_2$ 表示同层插入。 $x_1 > x_2$ 表示插入到最后一个孩子位置; $y_1 > y_2$,表示插入的节点或记录数据,在 $(2, 13)$ 和 $(1, 1)$ 之间。 $x_1 < x_2$ 表示插入到第一个孩子位置; $y_1 < y_2$ 表示插入 2 个记录之间^[9-11](如图 6 所示)。

由图可看出(a),(b),(d)插入表达式情况。图(c)中 X 将作为根 1 的最后一个孩子插入。表达式是: $X(2\ 3)(1\ 1)$ 。图(e)中插入 X 不属于增量更新,原来节点 3 是叶子节点,用 X 替换了节点 3 中的

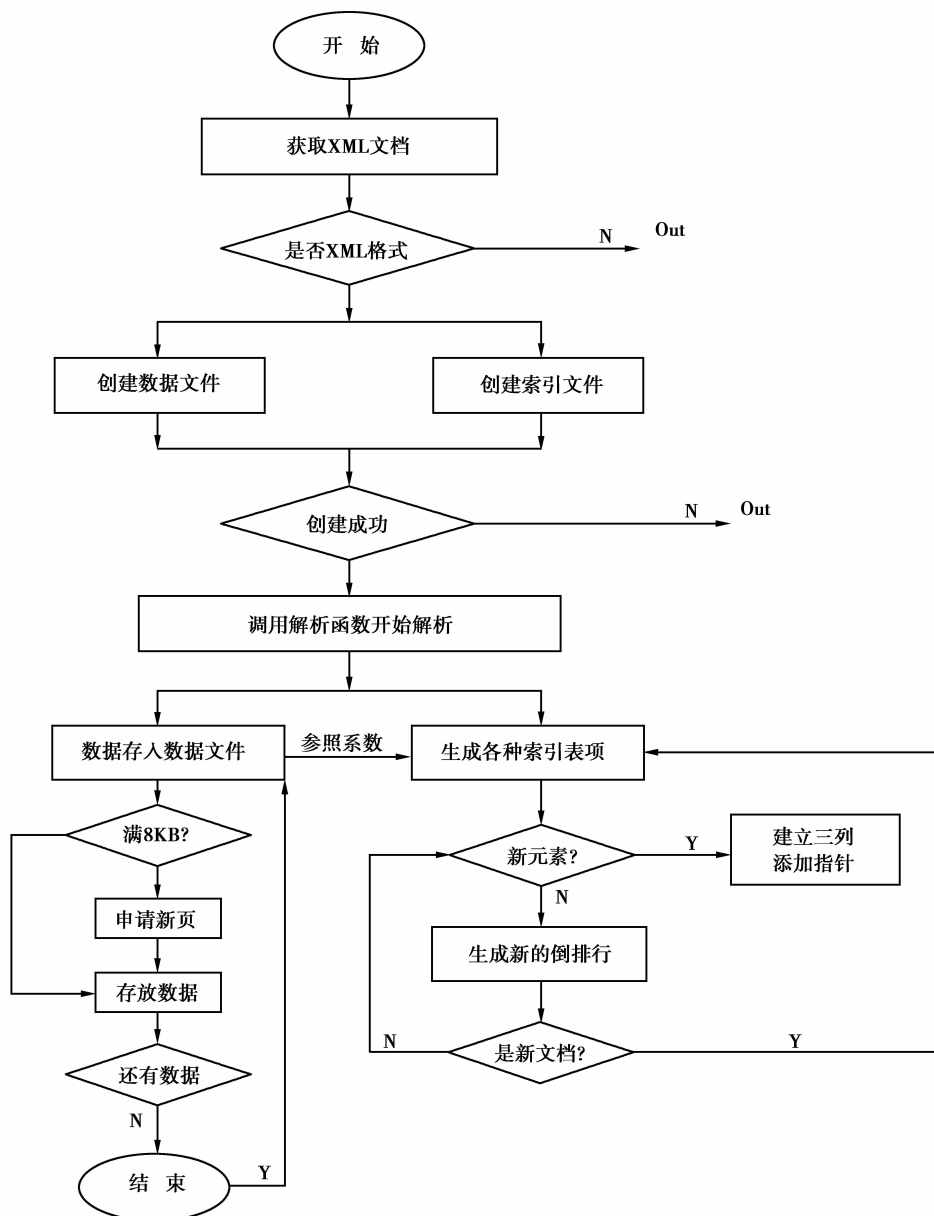


图 4 WNXD 存储流程图

值。更新索引文件,记录下更新记录的序数和层次,需要出现新的层次^[11-13]。

在更新增量数据时,如该页空间足够存储就直接写入,如果不够,就创建新页,添加索引,在数据表插入位置更改地址标记,将下一条指针指向刚插入的数据。

4 WNXD 功能实现

WNXD 系统功能主要包括存储 XML 数据、支持查询程序、增量更新程序。

4.1 WNXD 存储功能实现。

用 Button 触发加载 XML 文件,生成 WNXD 数据函数 CreateFile(),规定事件处理程序。

创建索引文件(ixl)(地址表、聚簇表)、数据文件(bxl)2 种空文件。规定文件头。创建 SAX 解析器,将事件处理程序分配给解析器,用解析函数 CWnxd()解析文件,将每个事件发送给处理程序,如图 5 所示。

引入 clock()函数计时。计算程序执行时间,用 (finish-start) /CLOCKS PER SEC,可以精确到 ms^[14-16]。将解析信息写入索引文件和数据文件如图 6 所示。

4.2 WNXD 查询功能实现。

WNXD 存储没有查询接口,但支持查询操作的功能实现。

要“查询更新”,需要打开“查询更新”界面,在查

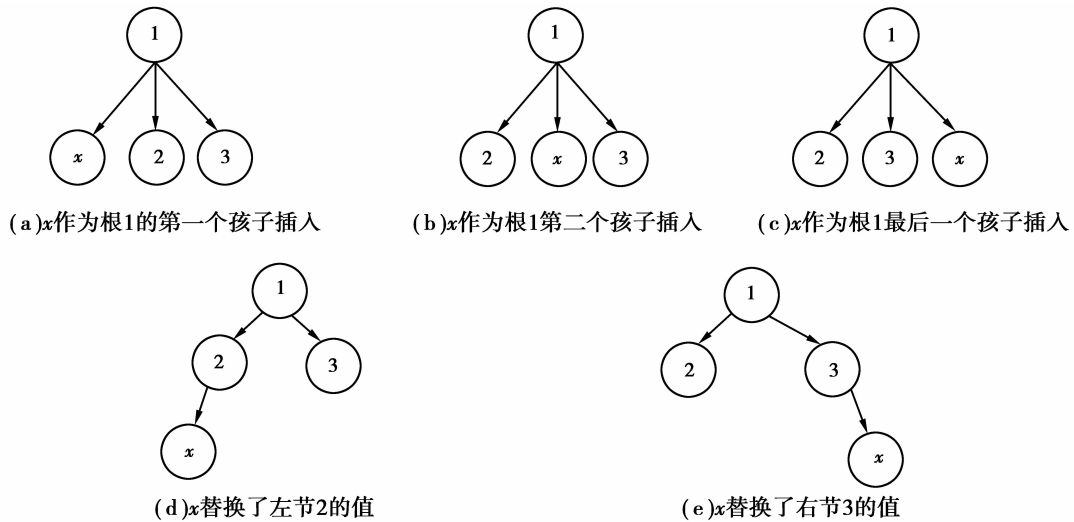


图 5 数据更新演示图

```

① SAXContentHandlerImpl()
② ParseURL(L" Wtn.xml" )
③ CWnxdDataFile * GetHead()
④ CMyNode * GetRoot()
⑤ void SaveToFile(CFile&file)
⑥ void SaveToIndexFile(CFile & file)
⑦ void SaveToDataFile(CFile & file)

```

图 6 存储解析算法



图 8 数据库存储

询框,把属性名或元素名等要查询的内容输入^[6],与属性或元素对应的三元组:顺序号,层号,页号,就会出现在显示框中。WNXD 查询子模块对处理高级 XPath, XQuery 语句支持更为良好。

4.3 WNXD 更新功能实现。

WNXD 支持对 XML 文档存储的增量更新,并能高效完成更新(如图 7、8 所示)。

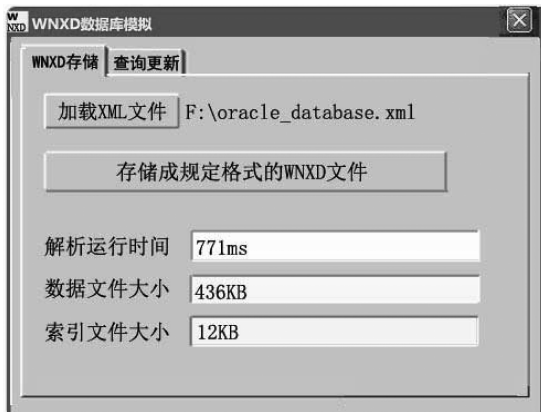


图 7 数据查询更新

```

<Metric NAME = "sessions_in_wait" TYPE
= "TABLE">
  <ValidIf>
    < Category Prop NAME = "
VersionCategory"
    CHOICES="pre8;8i;8iR2;9i;9iR2"/>
  </ValidIf>
  <Display>
    <Label NLSID="sessions_in_wait"
>System Sessions
    Waiting</Label>
  </Display>
</Metric>

```

wnxd. bxl 文件更新后,索引文件增大 125 字节,数据文件增大 778 字节。

4.4 存储时间和存储空间分析

经过模拟验证,分析 WNXD 存储空间和存储时间的比较(如图 9 所示)。

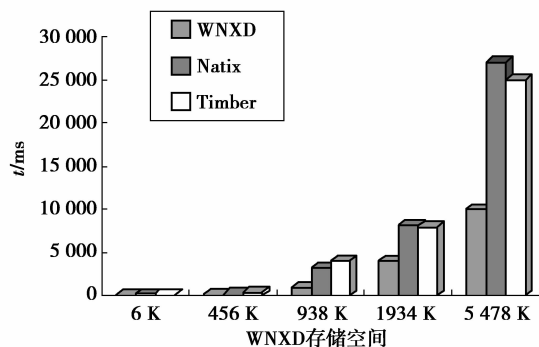


图 9 存储时间比较

通过对基于 SAX 解析的 3 种存储方案: WNXD, Natix, Timber 进行比较,发现在数据文件较小时,三种方案的存储时间相差不大。Natix 基于平衡树算法,当数据增加,树很大时,由于维护属性结构的文档映射,Natix 耗时增大,速度变慢。存储方案 Timber,面向对象数据库采用 shore,小对象处理时开销大,如果对象小而多时,存储速度就会变慢。WNXD 同时进行顺序解析、存储,存储时不用大量查询,存储速度接近解析速度,呈线性增长趋势。

存储空间的比较如图 10 所示。

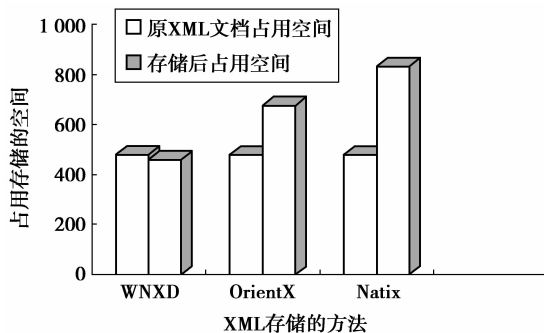


图 10 存储空间比较

选择 OrientX, Natix 进行对比试验,Natix 和 OrientX 在文件存储后,大小数据存储空间都有增长。WNXD 在文件存储后,不论大小数据,存储空间都比原来减小,存储优良,在后续的优化中有充分的扩展空间。

5 结 论

现在 XML 相关标准得以广泛推广应用,这样在 Web 上,就会有大量 XML 文档出现。在目前存储方法中,XML 数据存在支持查询效果差、存储性能低、数据更新慢的缺陷。出于有效管理的目的,需要把 XML 文档,通过一种新的方法存储到数据库

中。针对这些缺陷,深入研究了当前 Native XML 数据的存储方法,给出了一套 WNXD 文档存储策略,经模拟实验,这种新的半结构化信息的 WNXD 文档存储策略,对数据管理十分方便,查询更新效率得了提高。

参考文献:

- [1] 罗道峰,孟小峰. OrientStore: NativeXML 存储方法[J]. 计算机科学,2007,30(S1):105-110.
LUO DAO-FENG, MENG XIAO-FENG. OrientStore: native XML storage method [J]. Computer Science, 2007, 30(S1):105-110.
- [2] 余永平,朱卫东. NXD 研究与应用[J]. 现代计算机, 2004(1):31-34.
YU YONG-PING, ZHU WEI-DONG. NXD research and application [J]. Modern Computer, 2004 (1): 31-34.
- [3] 吕建华,王国仁,于戈. XML 数据的路径表达式查询优化技术[J]. 软件学报,2003,14(9): 1615-1620.
LV JIAN-HUA, WANG GUO-REN, YU GE. XML data path expression query optimization techniques [J]. Journal of Software, 2003, 14(9): 1615-1620.
- [4] 崔清华. XML 文档在关系数据库中的存储研究[J]. 微计算机信息,2009,23(24):165-168.
CUI QING-HUA. XML documents stored in relational database research [J]. Microcomputer Information, 2009,23(24):165-168.
- [5] 陈福生. Native-XML 数据库综述[J]. 计算机工程与设计,2004,25(6):932-935.
CHEN FU-SHENG. Native-XML database summary [J]. Computer Engineering and Design, 2004, 25 (6): 932-935.
- [6] DU W Y, LEE M L, LING T W. XML Structures for Relational Data [C] // Proceedings of the Second International Conference on Web Information Systems Engineering, Dec. 3-6, 2001. Kyoto, Japan: IEEE, 2001, 2:151-160.
- [7] TIAN F, DEWITT D J, CHEN J J, et al. The design and performance evaluation of alternative XML storage strategies[J]. ACM SIGMOD Record, 2007, 31 (1): 5-10.
- [8] CAREY M J, DEWITT D J, NAUGHTON J F, et al. The BULKY dbject-relational benchmark [J]. ACM SIGMOD Record, 1997, 26(2):12-21.
- [9] WIDOM J. Data management for XML: research directions[J]. IEEE Data EngBull, 1999,22(3),44-52.

- the inner-distance [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29 (2): 286-299.
- [7] ZHANG D S. Image retrieval based on shape [D]. Australia: Monash University, 2002.
- [8] LU G J, SAJJANHAR A. Region-based shape representation and similarity measure suitable for content-based image retrieval [J]. Multimedia Systems, 1999, 7(2): 165-174.
- [9] GOSHTASBY A. Description and discrimination of planar shapes using shape matrices [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1985, 7(6): 738-743.
- [10] SONKA M, HLAVAC V, BOYLE R. Image processing, analysis and machine vision [M]. United States: Cengage-Engineering, 2007.
- [11] BLUM H. A transformation for extracting new descriptors of shape; from models for the perception of speech and visual forms [M]. Cambridge: MIT Press, 1967: 362-380.
- [12] CHIO W P, LAM K M, SIU W C. Maximal disk based histogram for shape retrieval [C]// Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'03), April, 2003. Hong Kong, China: IEEE, 2003: 9-12.
- [13] ZHU S C, YUILLE A L. FORMS: a flexible object recognition and modeling system [J]. International Journal on Computer Vision, 1996, 20(3): 187-212.
- [14] SIDDIQI K, SHKOUFANDEH A, DICKINSON S, et al. Shock graphs and shape matching [J]. International Journal on Computer Vision, 1999, 35(1): 13-32.
- [15] SEBASTIAN T B, KLEIN P N, KIMIA B B. Recognition of shapes by editing their shock graphs [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(5): 550-571.
- [16] YANG S. Symbol recognition via statistical integration of pixel-level constraint histograms: a new descriptor [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2005, 27(2): 278-281.
- [17] BAI X, LATECKI L J, LIU W Y. Skeleton pruning by contour partitioning with discrete curve evolution [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007, 29(3): 449-462.
- [18] BAI X, LATECKI L J. Path similarity skeleton graph matching [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008, 30(7): 1282-1292.
- [19] 张恒博, 欧宗瑛. 一种基于色彩和灰度直方图的图像检索方法 [J]. 计算机工程, 2004, 30(10): 20-22. ZHANG HENG-BO, OU ZONG-YING. Method of content base image retrieval based on color and grey level histogram [J]. Computer Engineering, 2004, 30 (10): 20-22.
- [20] 任平红, 陈 鑫. 基于改进的边缘直方图的图像检索方法 [J]. 计算机技术与发展, 2007, 17(8): 183-186. REN PING-HONG, CHEN CHU. Methods of image retrieval based on improved edge histogram [J]. Computer Technology and Development, 2007, 17(8): 183-186.

(编辑 侯 湘)

~~~~~

(上接第 114 页)

- [10] LI Q Z, MOON B K. Indexing and querying XML data for regular path expressions [C]// Proceedings of the 27th International Conference on Very Large Data, September 11-14, 2001, Roma, Italy. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001: 361-370.
- [11] WANG G R, SUN B, LV J H, et al. RPE query processing and optimization techniques for XML databases [J]. Journal of Computer Science and Technology, 2006, 19(2): 224-237.
- [12] WANG J, MENG X F, WANG S. Integrating path index with value index for XML data [J]. Lecture Notes in Computer Science, 2003, 2642: 95-100.
- [13] TIAN F, DEWITT D J, CHEN J J, et al. The design and performance evaluation of alternative XML storage strategies [J]. ACM SIGMOD Record, 2002, 31(1): 5-10.
- [14] FLORESCU D, KOSSMANN D. Storing and querying XML data using an RDBMS [J]. Bulletin of the Technical Committee on Data Engineering, 1999, 22 (3): 27-34.
- [15] JMCHUGH J, ABITEBOUL S, GOLDMAN R, et al. Lore: a database management system for semistructured data [J]. SIGMOD Record, 2006, 26 (3): 54-56.

(编辑 侯 湘)